**Summary Report**

**(Interpreted) Requirements of the program**

* Read UML diagrams
* Construct codes with UML
* Create a class with 3 sub-classes
* Read user inputs to generate shapes
* Overriding computeArea() methods to generate correct area
* Sorting of shapes according to area in ASC and DSC order
* Polymorphism and function overloading

**Summary of implementation of each module in your program**

* Struct coord
* Class ShapeTwoD:public ShapeTwoD
  + Virtual toString() method
  + getArea() method
* Class Cross: public ShapeTwoD
  + Stringstream to return formatted output as required in assignment guideline
  + Virtual functions to check if points is inShape
  + Virtual function to check if point is onShape
  + computeArea() to override
* Class Circle: public ShapeTwoD
  + Stringstream to return formatted output as required in assignment guideline
  + Virtual functions to check if points is inShape
  + Virtual function to check if point is onShape
  + computeArea() to override
* Class Rectangle: public ShapeTwoD
  + Stringstream to return formatted output as required in assignment guideline
  + Virtual functions to check if points is inShape
  + Virtual function to check if point is onShape
  + computeArea() to override
* Class Square: public ShapeTwoD
  + Stringstream to return formatted output as required in assignment guideline
  + Virtual functions to check if points is inShape
  + Virtual function to check if point is onShape
  + computeArea() to override
* int Main()
  + call printMenu() to display main menu

**Reflections on program developments (E.g. assumptions made, difficulties faced, what could have been done better, possible enhancemenst in future, WHAT HAVE YOU LEARNT, etc)**

**Assumptions made**

* Calculation of area for the cross did not seemed to be as easy as explained in the class

**Difficulties faced**

* Understanding encapsulation and inheritance and how they work is not an easy task
* Unable to apply error handling when user enters gibberish input

**What could have been improved**

* The use of “try” and “catch” error handling could be implemented, as I have attempted but did not work out

**Possible enhancements**

* Implementation of error handling codes to make the program more user-friendly

**What I have learnt**

* Code in small paragraphs to troubleshoot in case there is any problems in compilation.